DETAILED ACTION

1. This Office Action is in response to the Request for Continued Examination filed 12/8/09. Claims 1-48, 60-62, 77, and 78. Claims 49-59, 63-76, and 79-81 are currently pending in the application.

Claim Objections

2. Claims 63, 64, 79, and 80 are objected to because of the following informalities: Claims 63 and 64 are objected to since they depend on canceled claim 61. Claims 79 and 80 are objected to since they depend on canceled claim 77. For purposes of examination, it will be assumed that claims 63 and 64 depend on claim 49 and claims 79 and 80 depend on claim 65. Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 49-55, 59, 65-71, 75, and 81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Li et al. (U.S. Patent 5,757,771) in view of Epps et al. (U.S. Patent US 6,721,316 B1) and in view of Winter et al. (U.S. Patent 5,778,414).

With respect to claims 49, 65, and 81, Li et al. discloses a system and method of managing packet transmission **(See the abstract and column 10 lines 7-13 of Li et al. for reference to a buffer management system and method including the transmission of ATM cells, which are packets)**. Li et al. also discloses receiving a plurality of packets wherein each packet has a header **(See column 6 lines 31-56 and claim 1 of Li et al. for reference to receiving a plurality of ATM cells, which are packets well known to include headers)**. Li et al. further discloses storing each received packet into one of a plurality of queues according to a quality of service parameter **(See column 2 line 66 to column 3 line14 and column 6 lines 31-56 of Li et al. for reference to storing each received ATM cells into one of a plurality of sub-queues according to a service class, which is a quality of service parameter, of each cell)**. Li et al. also discloses that each queue has a respective priority with one queue being a high priority queue **(See column 5 lines 29-65, column 10 lines 13-35, and Tables 1 and 2 of Li et al. for reference to each sub-queue having an output ranking that corresponds to a respective priority and for reference to one queue having the highest output ranking corresponding to a high priority queue)**. Li et al. further discloses that each queue other than the high priority queue has a corresponding timeout interval and whereupon expiration of a timeout interval will cause a packet stored in the queue corresponding to the timeout interval to be forwarded

ahead of packet stored in any other queue **(See column 10 lines 13-35, column 11 line 30 to column 12 line 61, Table 2, and Figure 3 of Li et al. for reference to each queue having a delay threshold, which is a timeout interval, whereupon expiration of the delay threshold will cause a cell stored in the sub-queue corresponding to the delay threshold to be dispatched ahead of packets stored in any other sub-queue)**. Li et al. also discloses forwarding each received packet to a network medium according to both priority of the queue in which the packet was stored and any expired timeout interval **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for reference to forwarding each cell according to both the output ranking and the delay thresholds of the queue in which the cell is stored)**. Li et al. further discloses dynamically allocating memory resources to each queue of the plurality of queues **(See column 6 line 57 to column 7 line 4 and column 7 lines 21-31 of Li et al. for reference to the buffer memory space allocated to each sub-queue being dynamically expanded and contracted based on size needs of the sub-queues)**. Li et al. does not specifically disclose the method being implemented as a computer readable media with instructions to causing a microprocessor to perform the method in a system where packets are received from a computer host memory. Li et al. also does not specifically disclose reading a quality of service parameter from the header of each received packet. Li et al. further does not specifically disclose maintaining both a list of free buffers and a list of used buffers for each queue with the lists comprising a plurality of pointers to memory locations that are available to store packets and a plurality of pointers to memory locations that are being used, respectively, wherein each pointer

has a size parameter specifying the size of the memory location indicated by the pointer.

With respect to claims 49, 65, and 81, Epps et al., in the field of communications, discloses a buffer management method being implemented as a computer readable media with instructions to causing a microprocessor to perform the method in a system where packets are received from a computer host memory **(See the abstract, column 7 lines 49-65, and column 44 lines 30-47 of Epps et al. for reference to a buffer management method implemented as a computer instructions stored in a computer readable medium executed by a CPU)**. Using a microprocessor to execute a buffer management method has the advantage of allowing the method to be implemented using software which is more easily adaptable than a method implemented using hard-wired instructions.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of Epps et al., to combine using a microprocessor to execute a buffer management method, as suggested by Epps et al., with the system and method of Li et al., with the motivation being to allow the method to be implemented using software which is more easily adaptable than a method implemented using hard-wired instructions.

With respect to claims 49, 65, and 81, Epps et al. discloses reading a quality of service parameter from the header of received packets **(See column 1 lines 12-30 and column 1 line 65 to column 2 line 20 of Epps et al. for reference to packet headers including quality of service parameters being read from packets)**. Reading a

quality of service parameter from the header of received packets has the advantage of allowing packets to be easily separated into difference classes for processing based on information explicitly contained within the packets.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of Epps et al., to combine reading a quality of service parameter from the header of received packets, as suggested by Epps et al., with the system and method of Li et al., with the motivation being to allow packets to be easily separated into difference classes for processing based on information explicitly contained within the packets.

With respect to claims 49, 65, and 81, Winter et al., in the field of communications, discloses a data processing system and method that maintains both a list of free buffers and a list of used buffers for queues with the lists comprising a plurality of pointers to memory locations that are available to store packets and a plurality of pointers to memory locations that are being used, respectively, wherein each pointer has a programmable size parameter specifying the size of the memory location indicated by the pointer **(See column 11 line 15 to column 13 line 3 of Winter et al. for reference to storing segment descriptor information, which corresponds to a pointer, for each memory segment, which corresponds to a buffer, including an indication of a segment's status as free or used, the programmable data length of the segment, and a pointer to the segment, with the segments being grouped into a linked list of free segments and used)**. Maintaining lists of pointers to free buffers

and pointers to used buffers has the advantage of providing a simple way to manage logical queues dynamically maintained in a shared memory.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of Winter et al., to combine maintaining lists of pointers to free buffers and pointers to used buffers, as suggested by Winter et al., with the system and method of Li et al. and Epps et al., with the motivation being to provide a simple way to manage logical queues dynamically maintained in a shared memory.

With respect to claims 50 and 68, Li et al. does not specifically disclose receiving packets being performed by a direct memory access download engine.

With respect to claims 50 and 68, Epps et al. discloses using direct memory access of data **(See column 14 lines 8-26 of Epps et al. for reference to using direct memory access)**. Using direct memory access has the advantage of reducing the amount of processor resources needed to transfer data between devices.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of Epps et al., to combine using direct memory access, as suggested by Epps et al., with the system and method of Li et al., with the motivation being to reduce the amount of processor resources needed to transfer data between devices.

With respect to claims 51 and 66, Li et al. discloses storing high priority packets into the high priority queue and storing low priority packets into a low priority queue **(See column 5 lines 29-65, column 10 lines 13-35, and Tables 1 and 2 of Li et al. for reference to storing higher priority cells in a sub-queue having a higher**

output ranking and storing lower priority cells in a sub-queue having a lower output ranking based on the service classes of the cells).

With respect to claims 52 and 67, Li et al. discloses storing high priority packets into the high priority queue, storing intermediate priority packets into an intermediate priority queue, and storing low priority packets into a low priority queue **(See column 5 lines 29-65, column 10 lines 13-35, and Tables 1 and 2 of Li et al. for reference to storing higher priority cells in a sub-queue having a higher output ranking, storing lower priority cells in a sub-queue having a lower output ranking, and storing intermediate priority cells in a sub-queue having an output ranking between the higher and lower output rankings based on the service classes of the cells)**.

With respect to claims 53 and 69, Li et al. discloses preempting packet forwarding from the high priority queue and forwarding a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for reference to preempting the dispatching of a cell from the highest priority sub-queue and dispatching a cell from a lower priority sub-queue when the delay threshold corresponding to the lower priority sub-queue has expired)**.

With respect to claims 54 and 70, Li et al. discloses preempting packet forwarding from one or more higher priority queues and forwarding a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for**

reference to preempting the dispatching of a cell from the higher priority sub-queues and dispatching a cell from a lower priority sub-queue when the delay threshold corresponding to the lower priority sub-queue has expired).

With respect to claims 55 and 71, Li et al. discloses forwarding a packet stored in the high priority queue when the high priority queue contains packets to be forwarded and no time interval has expired **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for reference to dispatching a cell from the highest priority sub-queue when the highest priority contains a cell and no delay threshold has expired in step 304)**. Li et al. also discloses preempting the high priority queue and forwarding a packet stored in an intermediate priority queue when the intermediate priority queue contains packets to be forwarded and the timeout interval corresponding to the intermediate priority queue has expired **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for reference to preempting the highest priority sub-queue from dispatching a cell and instead dispatching a cell stored in an intermediate priority sub-queue when the intermediate priority sub-queue contains cells to be forwarded and the delay threshold corresponding to the intermediate priority sub-queue has expired in steps 302 and 310)**. Li et al. further discloses preempting both the high priority queue and the intermediate priority queue and forwarding a packet stored in a low priority queue when the low priority queue contains packets to be forwarded and the timeout interval corresponding to the low priority queue has expired **(See column 11 line 30 to column 12 line 61 and Figure 3 of Li et al. for reference to preempting the highest priority sub-queue and the**

intermediate priority sub-queue from dispatching a cell and instead dispatching a cell stored in a low priority sub-queue when the low priority sub-queue contains cells to be forwarded and the delay threshold corresponding to the low priority sub-queue has expired in steps 302 and 310).

With respect to claims 59 and 75, Li et al. discloses the plurality of queues corresponding to a plurality of logical storage arrays wherein each logical storage array corresponds to one of the plurality of queues **(See column 2 line 66 to column 3 line 14 of Li et al. for reference to a single buffer being logically divided into logical storage arrays each corresponding to a sub-queue)**.

5. Claims 56, 57, 72, and 73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Li et al. in view of Epps et al. and Winter et al. and in further view of McAlpine (U.S. Publication US 2002/0141427 A1).

With respect to claims 56, 57, 72, and 73, the combination of Li et al., Epps et al. and Winter et al. does not specifically disclose received packets being compliant with the Ethernet protocol standard and the Infiniband protocol standard.

With respect to claims 56, 57, 72, and 73, McAlpine, in the field of communications, discloses a buffer management system and method using packets compliant with the Ethernet protocol standard and the Infiniband protocol standard **(See the abstract and page 1 paragraph 19 of McAlpine for reference to using packets compliant with Ethernet and Infiniband)**. Using packets compliant with the Ethernet protocol standard and the Infiniband protocol standard has the advantage of allowing a wider variety of standard packet types to be supported by the method.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of McAlpine, to combine using packets compliant with the Ethernet protocol standard and the Infiniband protocol standard, as suggested by McAlpine, with the system and method of Li et al., Epps et al., and Winter et al., with the motivation being to allow a wider variety of standard packet types to be supported by the method.

6. Claims 58 and 74 are rejected under 35 U.S.C. 103(a) as being unpatentable over Li et al. in view of Epps et al. and Winter et al. and in further view of Tezuka (U.S. Patent US 6,658,014 B1).

With respect to claims 58 and 74, the combination of Li et al., Epps et al., and Winter et al. does not specifically disclose the plurality of queues corresponding to a plurality of discrete storage arrays.

With respect to claims 58 and 74, Tezuka, in the field of communications, discloses a buffer management method using a plurality of discrete storage arrays corresponding to a plurality of queues **(See column 2 lines 40-46 and Figure 4 of Tezuka for reference to a packet buffer device using discrete buffers, which are discrete storage arrays corresponding to a plurality of queues)**. Using discrete storage arrays corresponding to queues has the advantage of isolating congestion of packets in the queues since an overflow of packets in one queue will not affect the amount of space available in other queues **(See column 2 lines 47-52 for reference to this advantage)**.

It would have been obvious for one of ordinary skill in the art at the time of the invention, when presented with the work of Tezuka, to combine using discrete storage arrays corresponding to queues, as suggested by Tezuka, with the system and method of Li et al., Epps et al., and Winter et al., with the motivation being to isolate congestion of packets in the queues since an overflow of packets in one queue will not affect the amount of space available in other queues.

Allowable Subject Matter

7. Claims 63, 64, 79, and 80 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

8. Applicant's arguments filed 12/8/09 have been fully considered but they are not persuasive.

Regarding Applicant’s argument that the data length information of Winter et al. refers to the size of a packet, not the size of a memory buffer, the Examiner respectfully disagrees. Winter et al. discloses conventional segment descriptor information including a segment's data length 12 (See column 11 lines 16-20 of Winter et al.).

Thus, it is the segment's data length and not a packet's data length, as argued, that is referred to by this information.

Regarding Applicant's argument that the data length is not a parameter of the pointer itself, as claimed, the Examiner respectfully disagrees. Applicant's claims include limitations regarding a pointer that has both information regarding a memory location and a size of the memory location. As shown in the rejections above, Winter et al. discloses segment descriptors that both size and locate block of memory including information comprising both a data pointer, which is a memory location, and a data length, which is a memory size (See column 11 lines 15-20 of Winter et al.). It the entire segment descriptor that of Winter et al. that has been mapped to the claimed pointer, in the rejections above, not just the data pointer. Thus, since the segment descriptor contains all the elements of the claimed pointer, it is equivalent to the claimed pointer.

Regarding Applicant's argument that Winter et al. does not disclose the size parameter being programmable to allow for buffers to vary in size, the Examiner respectfully disagrees. Winter et al. discloses convention segment descriptors containing information regarding a segment's data length (See column 11 lines 15-20 of Winter et al.). If different size segments were not to be used by conventional segment descriptors as disclosed by Winter et al., there would be no need for a segment descriptor to contain information regarding a segment's data length. Winter et al. further discloses a first embodiment whereby each segment is a fixed 256 byte size (See column 11 lines 35-38 of Winter et al.). Winter et al. also discloses that the 256 byte fixed size is a design configuration choice whereby other systems may use different

sizes (See column 11 lines 48-53 of Winter et al.). Winter et al. further discloses another embodiment where two different segment sizes, a 512 byte size and a 32 byte size, are used in conjunction with each other (See column 14 lines 24-53 of Winter et al.). Thus, Winter et al. teaches the use of different programmable segment sizes as necessitated by different requirements of data to be stored in the segments, and therefore, does disclose a programmable size parameter allowing for buffers to vary in size, as claimed.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON E. MATTIS whose telephone number is (571)272-3154. The examiner can normally be reached on M-F 8AM-5:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Huy Vu can be reached on (571)272-3155. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jason E Mattis
Primary Examiner
Art Unit 2461

JEM

/Jason E Mattis/
Primary Examiner, Art Unit 2461